Dual-Encoder Incongruity Transformer for Context-Aware Sarcasm Detection with Data Augmentation

Arti Verma
M.Tech Scholar
Department of Computer science & Engineering
Radharaman Engineering College
Bhopal, Madhya Pradesh, India
avartiverma3001@gmail.com

Rakesh Shivhare
Assistant Professor
Department of Computer science & Engineering
Radharaman Engineering College
Bhopal, Madhya Pradesh, India

Abstract: Sarcasm is hard for NLP systems because people rarely announce it directly—its meaning usually comes from how a sentence fits (or clashes) with what was said before. Many existing sarcasm detectors still rely on surface cues such as sentiment words, punctuation, or handcrafted lexical statistics, which often miss this "context vs. utterance" mismatch that drives sarcasm. In this paper, we introduce a Dual-Encoder Incongruity Transformer (DEIT) that treats sarcasm as an incongruity problem: the context and the target utterance are encoded separately using a shared Transformer, and their relationship is modeled using simple but effective interaction signals such as the absolute difference and element-wise product of their embeddings. Because sarcasm datasets are typically small, we also enlarge the training set to about 2000 samples through a hybrid augmentation strategy that mixes lightweight lexical edits with masked language model substitutions while keeping the class distribution balanced. On the evaluated dataset, DEIT achieves 98.33% accuracy, 98.33% F1-score, and an MCC of 0.9667, showing that explicitly modeling context–utterance incongruity leads to more reliable sarcasm detection.

Keywords: Sarcasm Detection, Context-Aware Classification, Transformer Models, Semantic Incongruity, Data Augmentation.

I. INTRODUCTION

Sarcasm is a tricky form of language where people often say the opposite of what they really mean. On microblogging platforms like Twitter, sarcasm shows up all the time: users write informally, bend grammar rules, and play with creative expressions. All of this makes it even harder to correctly understand the true sentiment behind a tweet. Traditional machine learning methods that rely mainly on surface patterns tend to miss these nuances, and as a result, sentiment or emotion analysis becomes unreliable whenever sarcasm is involved [1][2]. With recent progress in natural language processing and deep learning, however, we can now model context and meaning much more effectively, which has sparked renewed interest in treating sarcasm detection as an important research problem in its own right. Beyond classifying an utterance as sarcastic or non-sarcastic, emerging work also considers sarcasm in relation to its conversational context, speaker characteristics, and dialogue history, aiming to recognize not only whether sarcasm is present but also whether a given context is appropriate for sarcastic expression. This distinction is particularly relevant for building conversational AI systems that must decide when to interpret or generate sarcastic responses without degrading user experience [3][4]. Despite progress, several open challenges remain in Twitter sarcasm detection. First, datasets are typically small, noisy, and highly imbalanced, with sarcastic instances forming a minority class that is difficult to model [5][6][7]. Second, many state-of-the-art methods depend on large pretrained transformers and rich multimodal or user-profile information, which may not be available in practical deployments. Third, most existing systems focus on utterance-level sarcasm classification, while the problem of sarcasm-context detection—deciding whether a conversational context is suitable for sarcasm independently of a specific target utterance—has received limited attention. Motivated by these gaps, the present work investigates a context-driven approach to sarcasm detection on Twitter to improve robustness on short, noisy texts.

II. LITERATURE REVIEW

Early Twitter sarcasm and sentiment work relied on traditional machine-learning models built on hand-crafted lexical and syntactic features, where researchers like Riloff et al. [8] framed sarcasm as a clash between positive wording and negative context, and Prasanna et al. [9] showed that clause-level sentiment shifts and contrast patterns can help. Later, deep learning approaches (e.g., CNNs and RNNs) improved performance by learning subtler cues and sometimes using conversational context, with hybrid models like CAT-BiGRU combining convolution for local phrase patterns and attention-based BiGRU for longer dependencies, though they require enough labeled data and careful regularization to avoid overfitting. In parallel, researchers enriched feature sets

with signals like hyperbole, emotionally charged words, POS-based intensity, punctuation and capitalization, and Twitter-specific writing quirks, using tools like Brown clustering, dependency features, and chi-square selection to strengthen classic classifiers. More recent studies also highlight class imbalance as a key barrier and address it with embedding-based data augmentation and cost-sensitive methods like weighted Random Forests, both of which tend to produce more balanced and robust results on small, skewed datasets [1]. Early studies focused on classical supervised learning with rich manual features for text-based sarcasm detection on social networks. These works combine TF-IDF, Bag-of-Words, sentiment lexicons, punctuation cues, and Twitter-specific patterns with SVM, Naïve Bayes, Random Forest, and KNN, showing that carefully engineered feature sets significantly improve performance over plain n-grams. A later supervised approach grouped features into lexical, sarcasm-specific, and context-based categories and achieved about 90.5% accuracy with KNN when all three groups were combined, indicating that non-deep models remain competitive when supported by rich contextual and pragmatic features [10]. Diwan et al. [11] proposed a deep learning approa[ch for sarcasm detection on Twitter by combining pretrained word embeddings with convolutional and recurrent layers, achieving higher accuracy than classical machine learning baselines on both balanced and imbalanced datasets. They showed that automatically learned semantic and syntactic representations are more effective than purely handcrafted features, but still sensitive to data noise and class imbalance. Zhang et al. [12] proposed a tweet sarcasm detection model using a bidirectional GRU network with pooling to encode the full tweet sequence and compared it against a strong Bag-of-Words and n-gram baseline enriched with author history. Their experiments demonstrated that neural sequence representations and user-specific historical information substantially improve the detection of sarcastic usage patterns on Twitter. Alharbi and de Rijke [13] proposed a contextual-based sarcasm detection framework that jointly encodes each utterance and its surrounding conversational context into a compact representation. They reported that incorporating context increased F1 from roughly 49% (no context) to about 75% and reduced training time by 35.5%, confirming the central role of dialogue context for sarcasm understanding. Castro et al. [14] proposed the MUStARD corpus for multimodal sarcasm detection in dialogues, providing aligned text, audio, and visual data from TV shows with sarcasm annotations. Gandhi et al. [15] proposed a sarcasm detector on the MUStARD dataset by combining what was said with how it was said and what was shown on screen by fusing textual, audio, and visual cues using transformer-based attention. Alotaibi et al. [16] presented a broad review of sarcasm detection work on Twitter, spanning traditional machine learning methods, deep learning models, and hybrid techniques that is based on context-aware and transformer-based approaches. Ahmed et al. [17] proposed a supervised approach that organizes features into three intuitive groups and then tests multiple classifiers.

Table 1. Research Gaps and Its Solution in Proposed Approach

Research Gap	Reason	How the proposed model handled it
Shallow features don't capture sarcasm semantics	Existing approach relies on counts of sentiment words, POS tags, punctuation, caps, repetition, which miss context–utterance meaning mismatch	Proposed DEIT encodes context and utterance with a Transformer and models incongruity before classification
Weak modeling of context–utterance interaction	Baseline processes text without explicitly learning contradiction/irony between context and target utterance	Uses dual-encoder representations and interaction features, directly learning whether the utterance conflicts with the context
Augmentation introduces label noise	Synonym substitution can change the sarcasm intent producing unnatural samples	Uses contextual augmentation + controlled EDA that can apply self-filtering to reduce noise
Imbalance handled only at classifier level	Weighted RF helps class imbalance but cannot compensate for limited representation capacity of handcrafted features	Uses class-weighted loss in DEIT, while also learning richer representations that improve minority-class detection

III. METHODOLOGY USED

Sarcasm is hard for NLP because of mismatch between the utterance and its context. To tackle this, the paper proposed a Dual-Encoder Incongruity Transformer (DEIT) framework. This framework is presented in Fig. 1. To make training more robust and improve generalization augmentation is applied using lightweight lexical edits and contextual substitutions generated by a masked language model. This hybrid augmentation strategy reduces unwanted label drift.

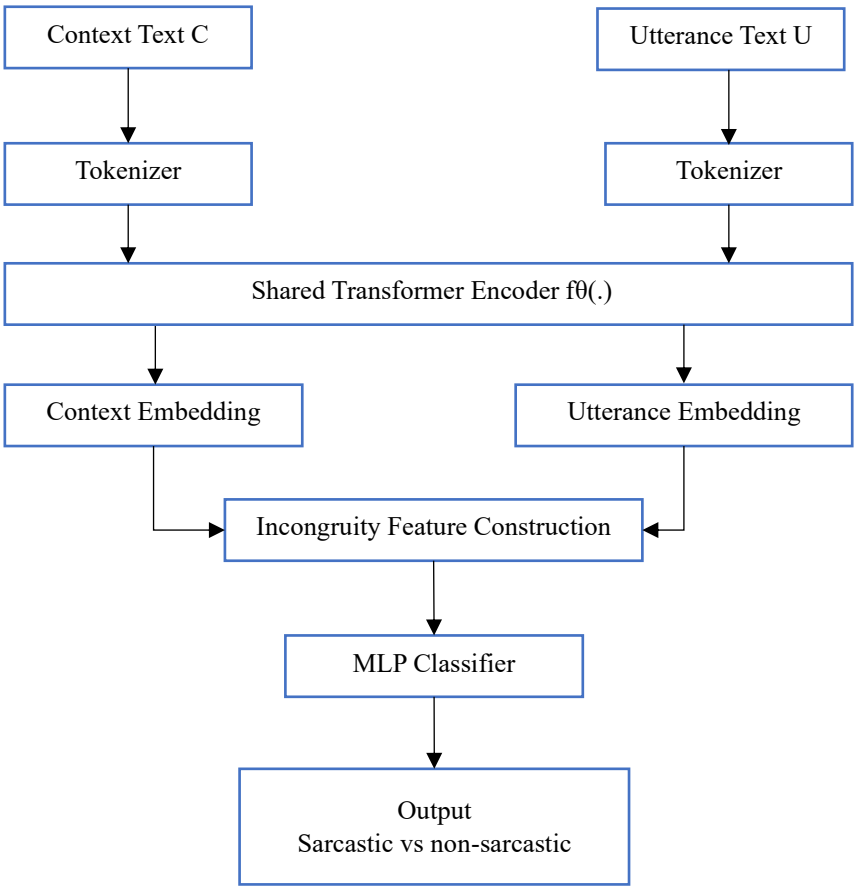

Fig. 1. Proposed Methodology

The methodology is organized as follows:

3.1 Data Preparation

The dataset contains conversational instances, where each instance includes:

- **Context**: a list of preceding utterances or conversational turns
- **Utterance**: the target sentence whose sarcasm label is to be predicted
- **Sarcasm label**: a binary value indicating sarcasm presence (1) or absence (0)

Each sample can be represented as:

$$x_i = (C_i, U_i) \qquad y_i \in \{0,1\} \qquad (1)$$

Where, C_i is the context text, U_i is the target utterance, and y_i is the sarcasm label.

To ensure fair evaluation and direct comparability with the baseline, we use a stratified train–test split. Stratification maintains the class distribution across splits, which is crucial for imbalanced datasets.

3.2 Preprocessing

Cleaning and Normalization: Although deep learning models can handle noisy text better than traditional methods, the dataset may contain irrelevant symbols, bracketed notes, non-ASCII characters, or extra whitespace. To reduce unnecessary noise, we apply minimal normalization:

- Convert text to lowercase
- Remove bracketed segments such as "[...]" or "(...)" if present
- Remove non-ASCII characters to standardize tokenization
- Normalize whitespace
- Let T be a raw input text. Preprocessing produces: $T' = normalize(T)$

Context Construction: Since context is often provided as a list of prior utterances, we concatenate them into a single text string: $C = c1 \parallel c2 \parallel \dots \parallel ck$

where $\parallel$ denotes concatenation with spaces. The final model input for each sample is a pair (C, U).

3.3 Data Augmentation to Expand Dataset Size

Deep models typically require a reasonable amount of data to generalize well. In sarcasm detection, small datasets lead to:

- High variance in results across splits

- Overfitting on dataset-specific cues
- Poor generalization to paraphrases or new expressions

Therefore, the proposed model expand the dataset size to approximately 3000 samples. However, sarcasm is highly sensitive to wording, so augmentation must be designed to preserve labels as much as possible.

Augmentation Strategy: The DEIT model use a hybrid augmentation pipeline combining:

- EDA-style lexical perturbations
- Contextual masked language model substitutions

The augmentation is applied to both: the target utterance U and optionally one randomly chosen context line in C. These yields varied but context-consistent examples.

EDA-Style Lexical Augmentations: EDA methods create new variants without generating entirely new sentences. The proposed method used:

- Replace a small fraction of words with WordNet synonyms.
  For a token sequence $w1, \dots, wn$ choose m eligible tokens and replace each with a synonym wj'.
- Then swap two random tokens a few times.
- Delete tokens with a small probability ppp while maintaining minimum length.
- These operations help the model learn robustness to small changes in word order and wording.

Contextual Substitution using Masked LM: EDA methods can be too crude and sometimes yield unnatural phrases. To generate more fluent variants, we use a masked language model (MLM) such as DistilRoBERTa. In this process:

- Select a word in the text (length ≥ 4, alphabetic)
- Replace it with a mask token <mask>
- Use the MLM to predict top-k candidates
- Replace the masked word with one predicted candidate

This tends to produce more natural substitutions because the replacement is conditioned on surrounding words.

To reduce label noise, an optional filtering strategy can be applied:

- Train a preliminary model on the original dataset
- Predict labels for augmented samples
- Keep only augmented samples where predicted label matches the original label
- This step helps discard augmented texts that accidentally changed sarcasm meaning.

Dual-Encoder Incongruity Transformer (DEIT)

Sarcasm often arises from a contrast between what is said (utterance) and what is implied by context. Therefore, modeling should focus on understanding context meaning, understanding utterance meaning and measuring the mismatch between them. Transformers (e.g., RoBERTa) provide strong contextual embeddings and capture semantic relations beyond surface word statistics.

Dual Encoder Representation: In DEIT, encode context and utterance are separately used a shared Transformer encoder $f(\cdot)$: $hC = f(C)$. where hC hU are dense vectors. Using a shared encoder reduces parameters and enforces consistent representation space.

Incongruity Feature Construction: To explicitly represent mismatch, the model compute interaction features by calculating:

- Absolute difference: $d = \mid hC - hU \mid$
- Element-wise product: $p = hC \odot hU$

Then we concatenate: $z = [hC; hU; d; p]$. where [;] denotes concatenation for hidden size H.

Classification Layer: The concatenated feature z is fed to a lightweight Multi-Layer Perceptron (MLP) using loss function such as class-weighted cross-entropy.

IV. RESULTS AND DISCUSSIONS

The proposed Dual-Encoder Incongruity Transformer (DEIT) was evaluated on the sarcasm dataset using a fixed stratified train–test split and the same evaluation parameters used for baseline comparison. Performance was measured using Accuracy, Precision, Recall, F1-score, and Matthews Correlation Coefficient (MCC).

The existing approach (Weighted Random Forest achieved an F1-score of 0.9747 with MCC = 0.9383. The proposed DEIT model achieved Accuracy = 0.9833, Precision = 0.9833, Recall = 0.9833, F1-score = 0.9833, and MCC = 0.9667. Therefore, this result shown in Fig 2 shows improvement of F1-Score of approx. 1% and MCC gain of 2.84%. These improvements indicate that the proposed approach not only increases overall classification effectiveness, but also yields a notably stronger correlation between predictions and ground truth across both classes.

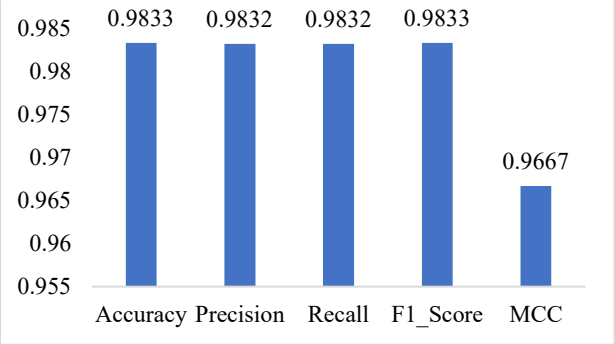

Fig. 2. Results Analysis of Proposed Methodology

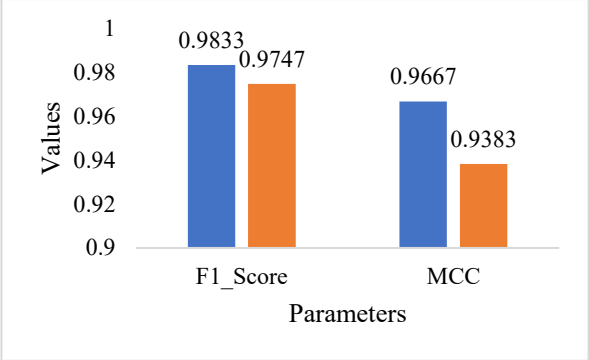

Fig. 3. Comparison Between Existing and Proposed Approach

Table 2. Conceptual Comparison Between Existing and Proposed Approach

Aspects	Existing Work	Proposed Work
Core idea	Sarcasm detection using handcrafted linguistic + sentiment + punctuation features with a traditional classifier	Sarcasm detection by modeling context–utterance semantic incongruity using Transformer embeddings
Input used	Primarily context text	Context + utterance
Feature representation	12 handcrafted features	Learned contextual embeddings + interaction features: [ctx, utt, |ctx−utt|, ctx⊙utt]
Model	Weighted Random Forest	Transformer encoder + MLP classifier
Handling class imbalance	Class-weighted RF	Class-weighted loss
Data augmentation	GloVe-based synonym replacement	EDA + masked-LM contextual substitution
Robustness to wording variation	Moderate	Higher
Training complexity	Low compute, fast training	Moderate compute

The baseline model depends on handcrafted surface-level features such as sentiment counts, POS counts, punctuation markers, capitalization, and repetition that are combined with a cost-sensitive random forest. This design can work well when sarcasm is expressed with overt signals such as excessive punctuation ("!!"), capitalization ("GREAT"), or strong polarity words. However, sarcasm frequently occurs without these cues and instead depends on the semantic contradiction between the conversational context and the literal utterance. DEIT explicitly addresses this by:

- Separately encoding context and utterance using a Transformer encoder, producing dense semantic embeddings.
- Computing interaction signals that model incongruity.
- Classifying sarcasm based on the combined representation, enabling the model to detect sarcasm even when surface cues are weak.

V. CONCLUSION

In this work, a context-aware sarcasm detection approach is proposed that is based on a Dual-Encoder Incongruity Transformer (DEIT). Instead of depending mainly on surface clues like punctuation, capitalization, or simple sentiment counts, the proposed method focuses on what actually drives sarcasm in many conversations by identifying the gap between the context and the literal utterance. By encoding the context and the utterance separately with a shared Transformer and then comparing them using simple interaction features, DEIT learns to recognize subtle sarcastic intent more reliably. To deal with the practical challenge of limited and imbalanced data, we also expanded the training set to around 3000 samples using balanced augmentation that combines light lexical edits with masked language model substitutions. The results show that this design leads to strong and consistent sarcasm classification under the same evaluation settings. In future,

this work can be strengthened by adding multimodal cues and testing robustness across different domains to ensure stable performance in real-world use.

Conflict of Interest: The corresponding author, on behalf of second author, confirms that there are no conflicts of interest to disclose.

References

[1] Senthilkumar, K. K., et al. "Twitter Sarcasm Detection using Natural Language Processing and Deep Learning Techniques." *2024 Global Conference on Communications and Information Technologies (GCCIT)*. IEEE, 2024.

[2] Pawar, Neha, and Sukhada Bhingarkar. "Machine learning based sarcasm detection on Twitter data." *2020 5th international conference on communication and electronics systems (ICCES)*. IEEE, 2020.

[3] Sarsam, Samer Muthana, et al. "Sarcasm detection using machine learning algorithms in Twitter: A systematic review." *International Journal of Market Research* 62.5 (2020): 578-598.

[4] Băroiu, Alexandru-Costin, and Ștefan Trăușan-Matu. "Automatic sarcasm detection: Systematic literature review." *Information* 13.8 (2022): 399.

[5] Alzaidi, Muhammad Swaileh A., et al. "Applied intelligence with deep learning assisted automated sarcasm recognition in twitter data." *Alexandria Engineering Journal* 128 (2025): 79-91.

[6] Alzaidi, Muhammad Swaileh A., et al. "Applied intelligence with deep learning assisted automated sarcasm recognition in twitter data." *Alexandria Engineering Journal* 128 (2025): 79-91.

[7] Helal, Nivin A., et al. "A contextual-based approach for sarcasm detection." *Scientific Reports* 14.1 (2024): 15415.

[8] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[9] M. S. Prasanna, S. G. Shaila, and A. Vadivel, "Polarity classification on Twitter data for classifying sarcasm using clause pattern for sentiment analysis," Multimedia Tools Appl., pp. 1–37, 2023.

[10] Alqahtani, Amal, Lubna Alhenaki, and Abeer Alsheddi. "Text-based sarcasm detection on social networks: A systematic review." International Journal of Advanced Computer Science and Applications 14.3 (2023).

[11] Diwan, A., & Narvekar, M. "A Deep Learning Approach for Sarcasm Detection on Twitter." International Journal of Information & Communication Technology Research, 2020.

[12] K. Veena and Dr. V. Sasirekha , Trans., "A Systematic Review of the Sarcasm Detection in the Twitter Dataset", IJRTE, vol. 12, no. 5, pp. 26–33, Jan. 2024, doi: 10.35940/ijrte.E7983.12050124.

[13] Alharbi, A., & de Rijke, M. "A Contextual-based Approach for Sarcasm Detection." PLOS ONE, 19(7), 2024.

[14] Sinha, Spriha, and M. Choudhary. "Sarcasm detection using deep learning approaches: A review." International Journal of Recent Technology and Engineering (IJRTE) 11.6 (2023): 50-58.

[15] Wang, Yufei, et al. "Multimodal Sarcasm Detection Based on MUStARD Dataset.", https://monagandhi09.github.io/asset/pdf/SarcasmDetector.pdf

[16] Alotaibi, F., et al. "A Systematic Review of the Sarcasm Detection in the Twitter Dataset." International Journal of Recent Technology and Engineering, 12(5), 2024.

[17] Abdullah Amer, Abdullah Yahya, and Tamanna Siddiqu. "A novel algorithm for sarcasm detection using supervised machine learning approach." AIMS Electronics & Electrical Engineering 6.4 (2022).